

## Modello del dominio di Erogazione/Fruizione Survey

Nel presente documento si riassumono i contenuti del deliverable "D2.1\_1 Modello di Rappresentazione di Unità Astratte di Interazione" relativo all'attività RI 2.1 Definizione di Modelli del Dominio di Erogazione/Fruizione di contenuti e servizi intelligenti, nell'ambito del secondo Obiettivo Realizzativo (OR 2) "Studio di servizi e contenuti intelligenti".

Il progetto SAPI richiede la modellazione di vari aspetti. La Figura 1 li contempla tutti ed in particolare evidenzia quelli inerenti il dominio di erogazione e fruizione dei servizi.

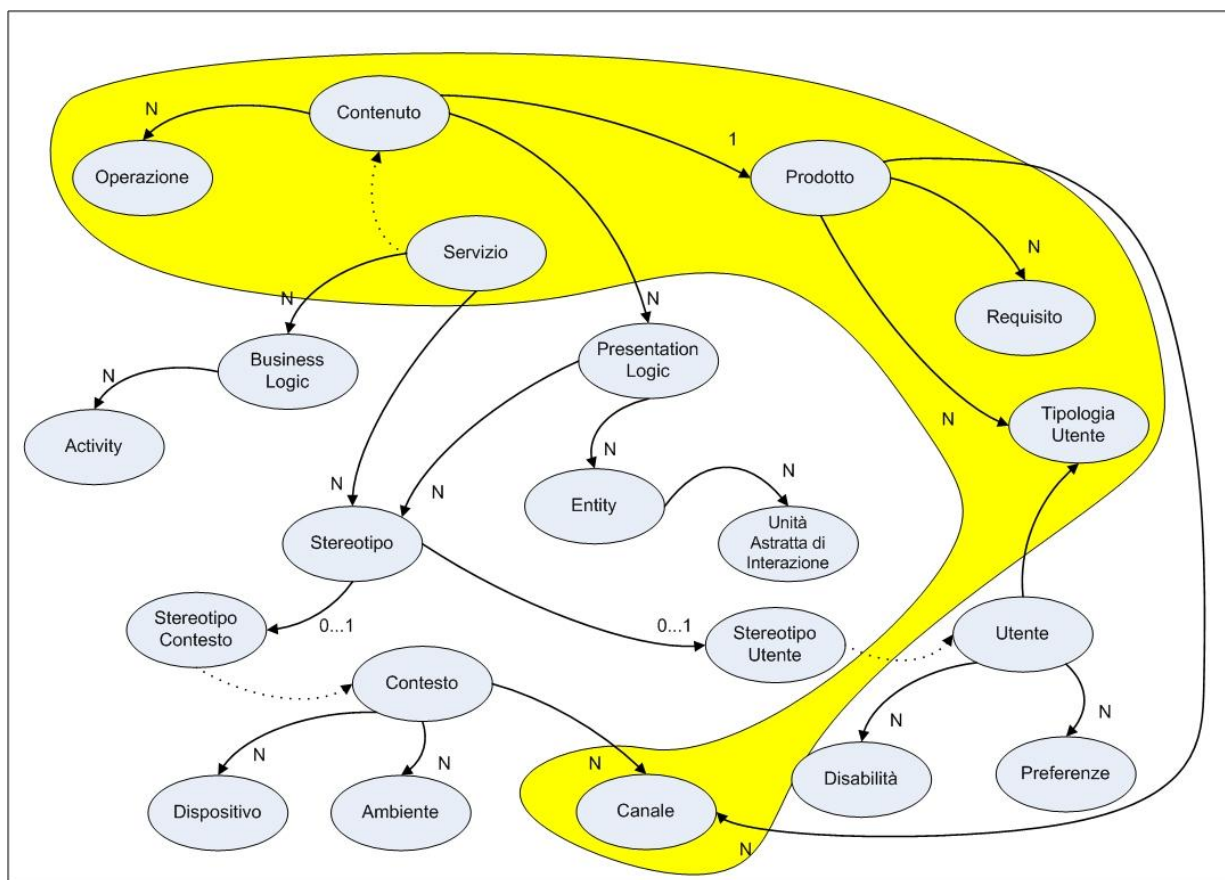


Figura 1: Vista d'insieme degli aspetti da modellare

Definire i modelli del dominio di SAPI prevede due fasi: una prima volta alla specifica formale ed alla rappresentazione del modello e una volta all'implementazione del modello stesso.

La necessità è quella di avere un modello condiviso, che faccia riferimento ad un lessico comune, che sistematizzi la conoscenza e garantisca la trasparenza semantica e l'interoperabilità tra sistemi, che risponda a requisiti di scalabilità e riusabilità e separi il dominio della conoscenza da quello operativo. Ciò posto la scelta è ricaduta sulle soluzioni ontologiche che sono caratterizzate dalla presenza di classi e relazioni. Le prime identificano concetti rilevanti nel dominio, mentre le relazioni, o proprietà, identificano il legame tra le classi che costituiscono soggetto ed oggetto della relazione. La terminologia utilizzata per riferire questi aspetti è la seguente:

Classe --- Relazione --> Classe.

Per la definizione del modello del dominio di erogazione e fruizione dei servizi si è definito prima un modello astratto del contesto in cui nasce e si evolve SAPI, Poste Italiane S.p.A.; poi si è definito e rappresentato in un opportuno linguaggio il modello di SAPI.

La Figura 2 riporta il modello di Poste Italiane S.p.A..

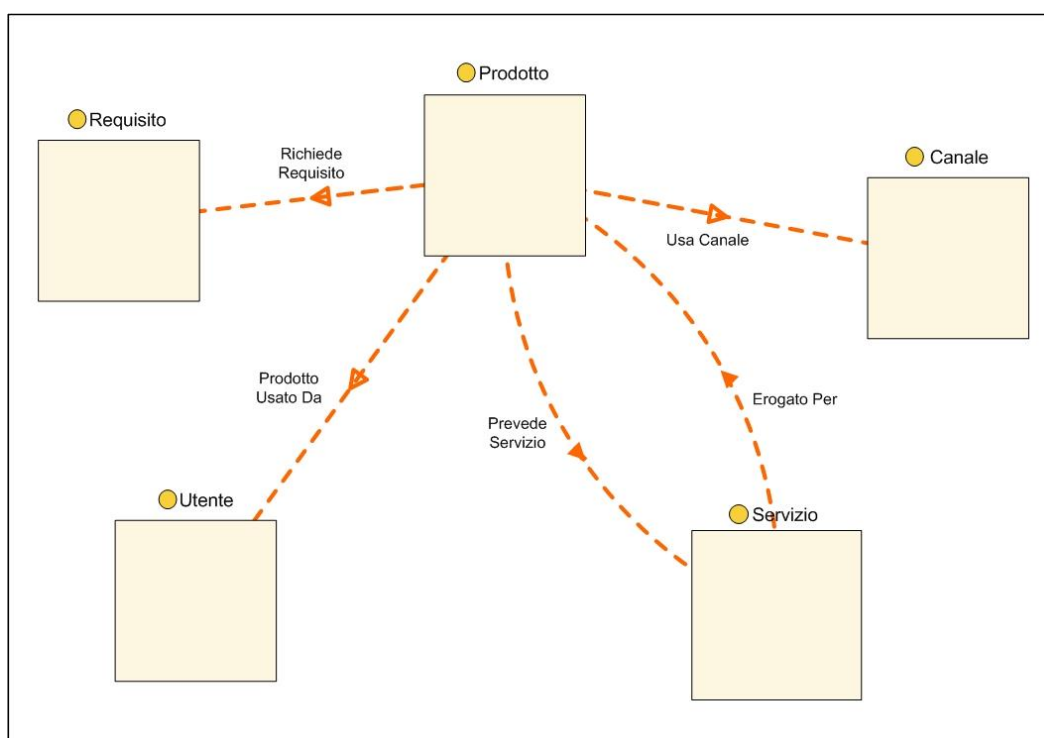


Figura 2: Modello del dominio di erogazione e fruizione dei prodotti e dei servizi di Poste Italiane S.p.A.

I concetti cardine del dominio individuati sono i seguenti.

1. **Prodotto:** qualsiasi entità che può, ma non per forza deve, costituire elemento di profitto.
2. **Servizio:** operazione atomica elementare che può essere messa a disposizione da più prodotti ed eseguita su più prodotti.
3. **Canale:** mezzo attraverso il quale si utilizza un particolare prodotto e viene richiesta l'esecuzione di un particolare servizio.

4. **Requisito:** ciò di cui si ha bisogno perché si possa eseguire un servizio che un prodotto mette a disposizione.
5. **Utente:** entità giuridicamente identificata che usufruisce dei prodotti di Poste Italiane S.p.A. e dei servizi da questi messi a disposizione..

Le relazioni tra le classi, invece, sono le seguenti.

1. **Prevede Servizio:** indica quale servizio mette a disposizione un prodotto.
2. **Erogato Per:** indica per quale prodotto è erogato un servizio.
3. **Richiede Requisito:** indica i requisiti necessari perché si possa usufruire di un prodotto.
4. **Prodotto Usato Da:** lega l'utente ai prodotti che può utilizzare.
5. **Usa Canale:** modella il fatto che l'Utente utilizza un prodotto ed i servizi che esso mette a disposizione attraverso un canale.

Dal momento che tra gli scopi dell'Obiettivo Realizzativo 5 (OR5) c'è quello di realizzare un prototipo per l'erogazione di un servizio per cui in Figura 3 è riportato un esempio reale con le istanze relativamente al pagamento bollettino.

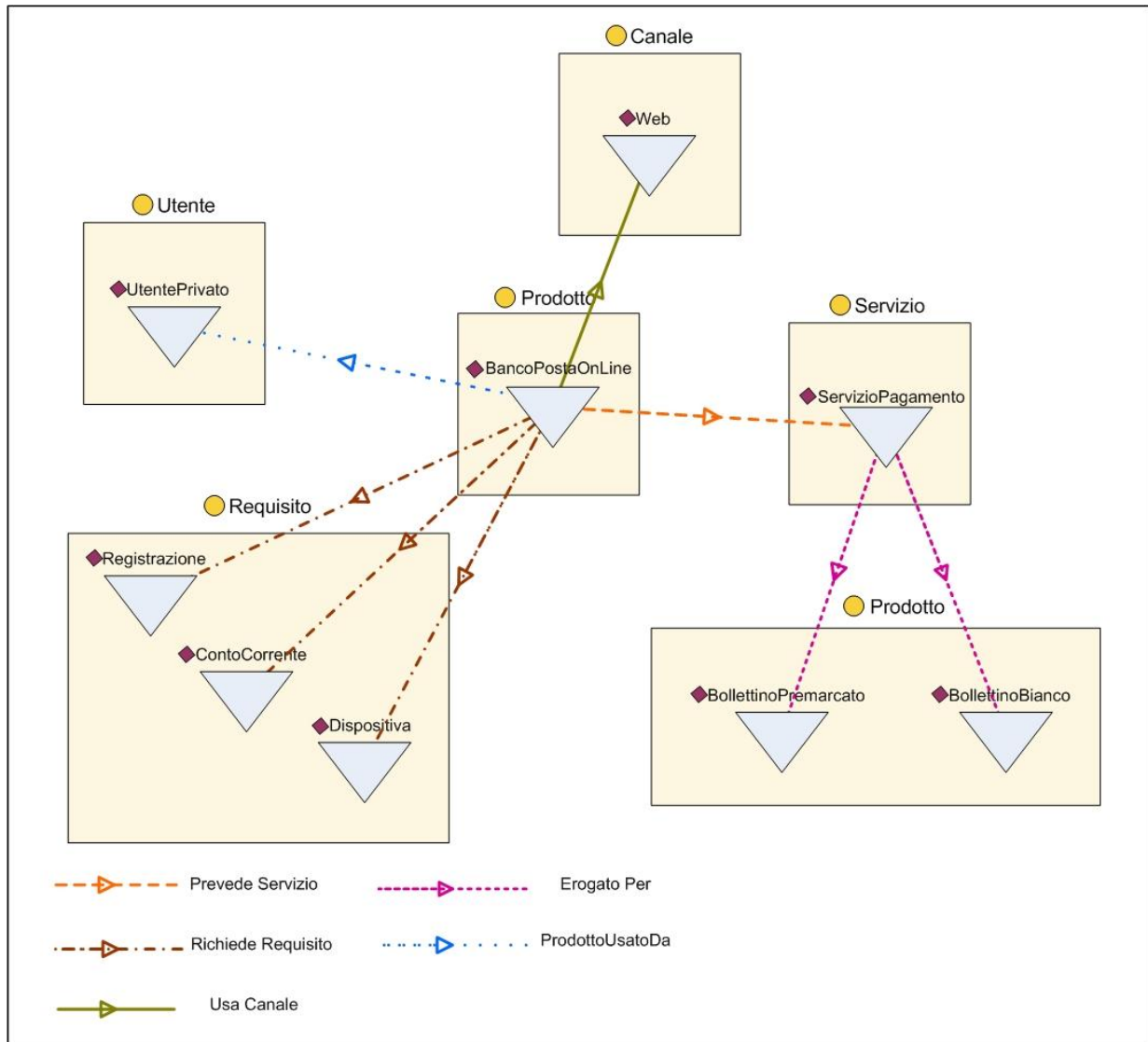


Figura 3: Modello del Pagamento Bollettino

A partire da questo modello si deriva quello di SAPI che viene riportato in Figura 4.

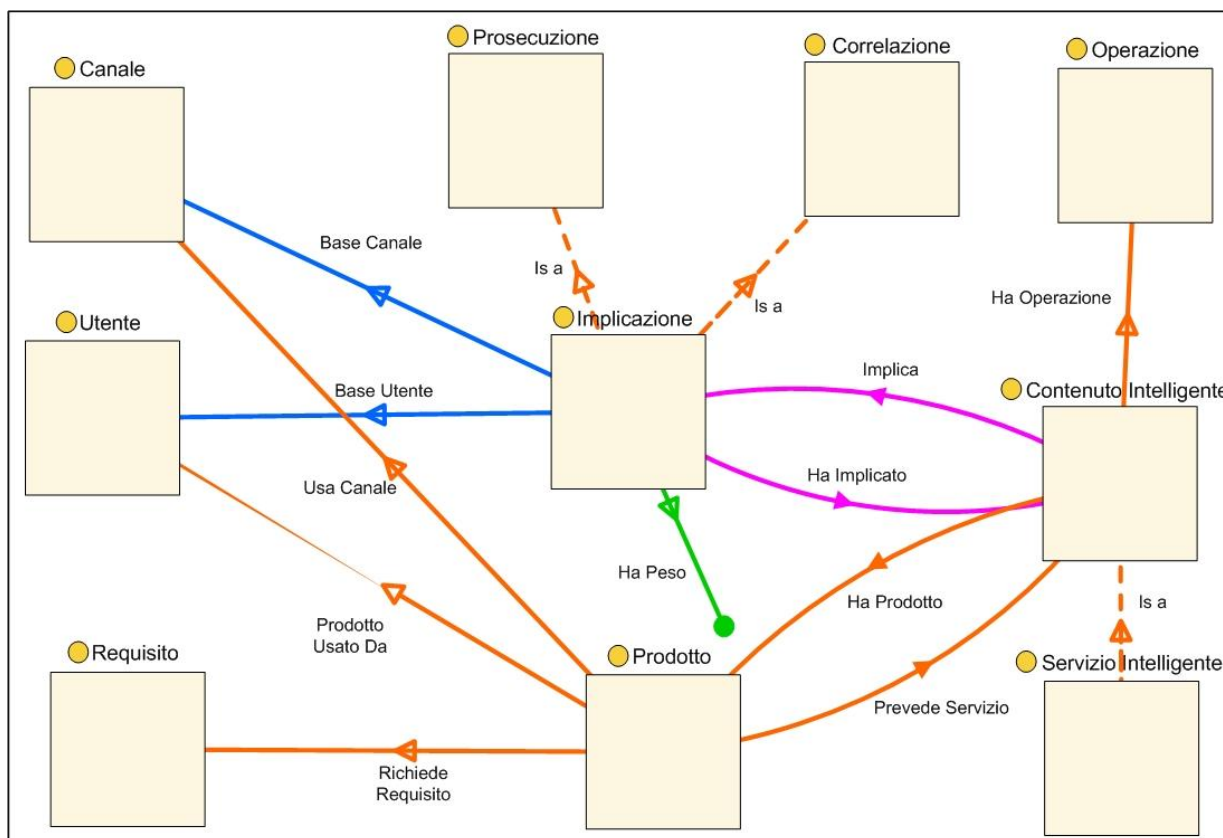


Figura 4: Modello del dominio di erogazione e fruizione dei prodotti e dei servizi di SAPI

Rispetto al modello precedente i concetti che restano inalterati sono quelli di **Prodotto**, **Requisito**, **Canale** e **Utente**. Costituisce elemento di novità, invece, il concetto di servizio che porta alla definizione di tre nuovi concetti.

1. **Operazione**: coincide con il concetto di Servizio definito nel modello precedente.
2. **Servizio Intelligente**: è la composizione di un'Operazione e di un Prodotto.
3. **Contenuto Intelligente**: concetto padre del Servizio Intelligente rispetto ad esso presenterà le stesse capacità di adattamento ma una logica di business elementare.

L'aggiunta di nuove entità ha portato anche ad una nuova configurazione delle relazioni. Pertanto, mentre sono rimaste inalterate **Usa Canale**, **Richiede Requisito**, **Prodotto Usato Da** e **Prevede Servizio**, la proprietà **Erogato Per** è andata persa e sono state definite le seguenti.

1. **Ha Prodotto**: indica per quale prodotto è erogato un contenuto intelligente.
2. **Ha Operazione**: indica da quali operazioni è composto un contenuto intelligente..

A questi elementi, infine, vanno aggiunti altri che sono derivati dalla specifica di relazioni implicite ossia calcolate sulla base delle interazioni dell'utente. Quello che si può verificare, infatti, è che un utente abbia delle preferenze in merito alla sequenza di servizi di cui usufruisce, o in merito al canale mediante il quale usufruisce di un particolare servizio.

Ciò ha portato alla definizione di tre nuovi concetti:

1. **Implicazione**: modella il concetto di relazione implicita e costituisce un tramite tra due contenuti intelligenti;

2. **Correlazione:** concetto figlio di Implicazione, indica la correlazione che esiste tra due contenuti intelligenti;
3. **Prosecuzione:** concetto figlio di Implicazione, indica con quale sequenza vengono fruiti i servizi.

Di conseguenza è stato necessario anche definire delle nuove relazioni:

1. **Implica:** indica quale implicazione è associata ad un particolare Contenuto Intelligente;
2. **Ha Implicato:** indica quale è il contenuto destinazione dell'implicazione;
3. **Base Canale:** indica che l'implicazione tra contenuti intelligenti è calcolata sulla base del canale;
4. **Base Utente:** indica che l'implicazione tra contenuti intelligenti è calcolata sulla base dell'utente;
5. **Ha Peso:** indica il peso che assume un'implicazione.

Per avere un'idea di come funziona una relazione implicita si possono considerare due istanze della classe Servizio Intelligente (Pagamento Bollettino e Ricarica PostePay). Il sistema sarà in grado di intuire quale è la correlazione tra i due servizi sulla base di quante volte l'utente fa seguire la fruizione del servizio Pagamento Bollettino al servizio di Ricarica PostePay.

Apparentemente le due relazioni Correlato A e Segue non hanno differenze. La distinzione però deve essere individuata nelle modalità di calcolo di queste stesse.

Prima di indicare dove sta la differenza si illustra la metodologia per il calcolo delle relazioni implicite. A tal proposito si fa riferimento alla regola di Hebb secondo cui neuroni che sono attivati simultaneamente (co-attivati) diventano più strettamente associati. Il rafforzamento dell'associazione è proporzionale al grado di attivazione di ciascun neurone  $A(i,t)$ . Il grado di co-attivazione, d'altra parte, decade esponenzialmente con il crescere dell'intervallo di tempo intercorrente tra l'attivazione di  $i$  e l'attivazione di  $j$  rappresentato con  $(t_j - t_i)$ .

Ciascun episodio di co-attivazione porta quindi all'aggiornamento della matrice di attivazione attraverso la formula seguente:

$$a_{ij} = a_{ij} + \delta_{ij} \quad \text{dove} \quad \delta_{ij} = A(i, t_i) \cdot A(j, t_j) \cdot \exp(-k \cdot (t_j - t_i))$$

Dove il coefficiente  $k > 0$  regola la velocità di attenuazione nel tempo dell'impulso di attivazione del neurone.

In questo caso, piuttosto che le possibilità di co-attivazione tra i neuroni del cervello, interessano le possibilità di co-utilizzo di servizi diversi da parte dello stesso utente ovvero di attivazione futura di un servizio  $j$  da parte di un utente sapendo che attualmente ha attivato il servizio  $i$ . In questo caso, ogni componente  $a_{ij}$  della matrice di attivazione indicherà il grado di possibilità di fruizione del servizio  $j$  seguita dalla fruizione del servizio  $i$ .

Le associazioni, pertanto, sono apprese attraverso la formula seguente:

$$a_{ij} = a_{ij} + \delta_{ij} \quad \text{dove} \quad \delta_{ij} = \exp(-k \cdot (t_j - t_i))$$

Dove per semplicità all'attivazione di ciascun servizio da parte dell'utente si associa un valore pari a 1 ovvero  $A(i, t_i) = A(j, t_j) = 1$ . Supponendo che la sequenza degli ultimi  $n$  servizi acceduti da parte di un determinato utente al tempo  $t$  (a parte il corrente) sia la seguente:

$$(S(t-n), \dots, S(t-2), S(t-1)) \quad \text{allora} \quad \delta_{S(t-i)S(t)} = \exp(-k \cdot i) \quad \text{per } i = 1 \dots n.$$

In tal senso si semplifica ulteriormente il calcolo considerando, per ogni utente, solo gli ultimi  $n$  servizi acceduti e non tutta la storia.

Per tornare alla differenza tra la relazione implicita di correlazione e quella di prosecuzione ora si può dire che ciò che distinguerà le due relazioni sarà il coefficiente  $K$ . Infatti il coefficiente  $K_C$  definito per la relazione Correlato A assume un valore più alto rispetto al coefficiente  $K_S$  definito per la relazione Segue. Di conseguenza la relazione Correlato A rappresenta un legame lasco tra i due servizi, in quanto l'utente non li fruisce in rapida successione, mentre la relazione Segue rappresenta un legame stretto, in quanto l'utente fruisce dei servizi correlati in rapida sequenza.

Per la specifica di questo modello si è scelto OWL che presenta:

- a. una sintassi ben definita;
- b. una semantica ben definita;
- c. il supporto al ragionamento;
- d. elevata potenza espressiva.

In commercio sono presenti anche dei tool di editing che consentono la specifica di modelli ontologici. Di tutti quelli esistenti è stata fatta una valutazione accurata di quelli più utilizzati: Protège e TopBraidComposer. La soluzione più vantaggiosa sembra essere la seconda.

Allo stesso modo è stata fatta una valutazione in merito agli strumenti di persistenza Sesame e Jena entrambe progetti Open Source.

Infine, dal momento che le informazioni qui modellate devono dare adito a deduzioni è stata fatta anche una valutazione degli strumenti di ragionamento attualmente esistenti e più utilizzati.

Jena ne presenta uno, ma non è molto avanzato. A questo vanno aggiunti i tre più utilizzati in letteratura che sono RacerPro, Pellet e Fact++. Pellet, sebbene dal punto di vista prestazionale non sembra essere molto vantaggioso, risulta molto utilizzato tanto che è anche la soluzione integrate nell'editor TopBraid.